



BuR – Business Research
Official Open Access Journal of VHB
German Academic Association for Business Research (VHB)
Volume 6 | Issue 1 | May 2013 | 55–75

A Column-Generation Approach for a Short-Term Production Planning Problem in Closed-Loop Supply Chains

Florian Sahling, Institute of Production Management, Leibniz Universität Hannover, Germany, E-mail: florian.sahling@prod.uni-hannover.de

Abstract

We present a new model formulation for a multi-product lot-sizing problem with product returns and remanufacturing subject to a capacity constraint. The given external demand of the products has to be satisfied by remanufactured or newly produced goods. The objective is to determine a feasible production plan, which minimizes production, holding, and setup costs. As the LP relaxation of a model formulation based on the well-known CLSP leads to very poor lower bounds, we propose a column-generation approach to determine tighter bounds. The lower bound obtained by column generation can be easily transferred into a feasible solution by a truncated branch-and-bound approach using CPLEX. The results of an extensive numerical study show the high solution quality of the proposed solution approach.

JEL-classification: C61, M11

Keywords: closed-loop supply chains, remanufacturing, column-generation

Manuscript received April 12, 2012, accepted by Karl Inderfurth (Operations and Information Systems), October 24, 2012.

1 Introduction

In the last decade, many researchers have been attracted by production planning aspects in closed-loop supply chains (Fleischmann, Bloemhof Ruwaard, Dekker, van der Laan, van Nunen, and van Wassenhove 1997; Guide and Van Wassenhove 2001; Rubio, Chamorro, and Miranda 2008; Sbihi and Eglese 2010). In addition to forward-oriented supply chains from the manufacturer to the customer, the reverse direction also has to be taken into account in the field of closed-loop supply chains. This means that customers return old and used products to the manufacturer at the product's end-of-use. Well-known examples are printer cartridges and single-use cameras (Akcali and Cetinkaya 2011: 2376).

These returned products can be remanufactured to become products, which are as good as new. Therefore, remanufactured products can also be used to fulfill customers' demand. Returned products are

denoted as recoverables, i.e., these items have to be remanufactured. After (re)manufacturing products are denoted as serviceables. The products can be (re)manufactured on a capacity-restricted production system. For (re)manufacturing products a setup is necessary, which causes setup costs and/or setup times. Hence, a multi-product capacitated lot-sizing problem arises, as the respective products compete for the production system's scarce capacity. Most of the established model formulations neglect either the capacity constraints or the interactions of multiple products. Therefore, a new model formulation for the multi-product capacitated lot-sizing problem with product returns and remanufacturing (CLSP-RM) is presented. Furthermore, a solution approach by combining column generation and a truncated branch-and-bound approach is proposed.

The remainder of this paper is organized as follows: In section 2 we provide a literature review of lot-sizing problems in closed-loop supply chains. Afterwards, we present a new model formulation

for a capacitated lot-sizing problem with remanufacturing in section 3. In this section, we also introduce valid inequalities to tighten the lower bounds of the LP relaxation. In section 4 a column-generation approach is proposed to determine a lower bound close to the optimal solution. To generate a feasible production plan based on this lower bound, a truncated branch-and-bound method is applied. Numerical results are reported in section 5. The paper ends with a summary and an outlook for further research in section 6.

2 Literature Review

Reviews of the literature on dynamic lot sizing in general are given, e.g., by [Karimi, Fatemi Ghomi, and Wilson \(2003\)](#), [Quadt and Kuhn \(2008\)](#) and [Buschkuhl, Sahling, Helber, and Tempelmeier \(2010\)](#). In this section, we focus only on dynamic lot-sizing problems with product returns and remanufacturing. However, the literature on lot-sizing problems in closed-loop supply chains is rather limited.

One of the first papers on dynamic lot-sizing problems in closed-loop supply chains was presented by [Richter and Sombrutzki \(2000\)](#). They suggested a model formulation for the reverse Wagner-Whitin problem ([Wagner and Whitin 1958](#)) by assuming that the demand can be totally satisfied by remanufacturing returned products. Richter and Weber (2001) extended the reverse Wagner-Whitin problem by introducing variable manufacturing and remanufacturing costs. [Richter and Sombrutzki \(2000\)](#) as well as [Richter and Weber \(2001\)](#) proved that the reverse Wagner-Whitin problem can usually be solved in polynomial time.

[Teunter, Bayindir, and van den Heuvel \(2006\)](#) described model formulations for the single-level uncapacitated lot-sizing problem with remanufacturing (SLULSP-RM) based on the Wagner-Whitin problem. Two cases are distinguished: In the first case, the products are manufactured and remanufactured on the same (uncapacitated) production system and only one joint setup is necessary for both remanufacturing and manufacturing. In the second case, it is assumed that manufacturing and remanufacturing take place on different (uncapacitated) production systems. Thus, a separate setup is required on each production system.

[Teunter, Bayindir, and van den Heuvel \(2006\)](#) proved that in the case of joint setups the SLULSP-

RM can be solved in polynomial time using an adapted version of [Wagner and Whitin's](#) dynamic programming approach. Furthermore, [van den Heuvel \(2004\)](#) showed that the SLULSP-RM is \mathcal{NP} -hard in the case of separate setups.

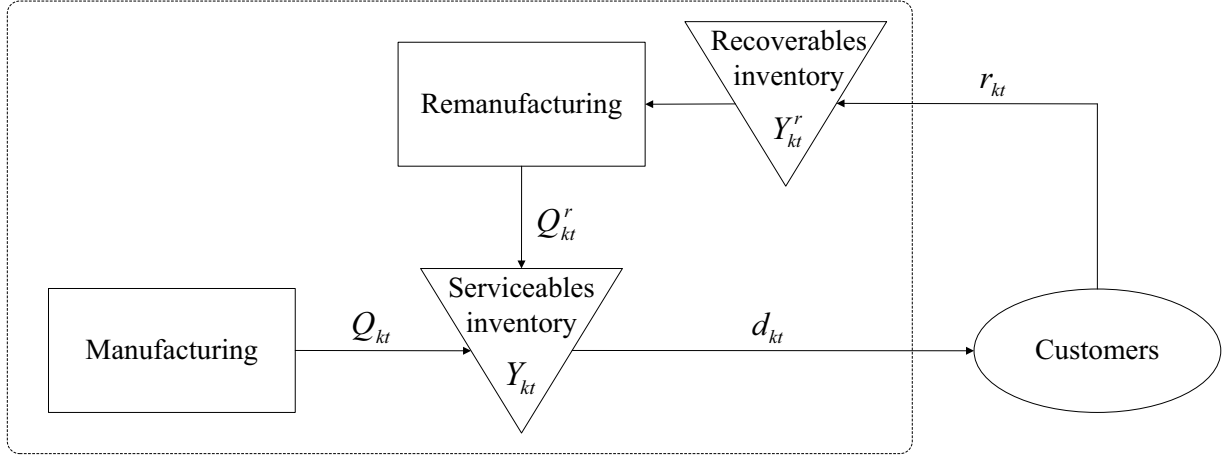
To solve both cases of the SLULSP-RM, [Teunter, Bayindir, and van den Heuvel \(2006\)](#) adapted well-known heuristics for the Wagner-Whitin problem, namely the least-unit cost approach, the Silver-Meal heuristic and the part-period-balancing approach. They pointed out that the adapted Silver-Meal heuristic outperforms the other approaches. [Schulz \(2011\)](#) introduced further variants of the Silver-Meal heuristic to solve the SLULSP-RM with separate setup costs. Furthermore, he proposed improvement steps yielding a higher solution quality than the Silver-Meal heuristic alone.

To strengthen the lower bound of the LP relaxation of the SLULSP-RM, [Retel Helmrich, Jans, van den Heuvel, and Wagelmans \(2010\)](#) adapted valid inequalities proposed by [Barany, van Roy, and Wolsey \(1984\)](#) for the Wagner-Whitin problem and present reformulations based on the shortest-path problem for the SLULSP-RM.

[Quariguasi Frota Neto, Walther, Bloemhof, van Nunen, and Spengler \(2009\)](#) integrated sustainable aspects into the SLULSP-RM by taking the required energy for (re)manufacturing into account. The SLULSP-RM with joint setups was extended by [Schwarz, Buscher, and Rudert \(2009\)](#) and [Schwarz \(2010\)](#) by allowing disposals of the returned products. The dynamic programming approach by [Teunter, Bayindir, and van den Heuvel \(2006\)](#) was adapted to solve the SLULSP-RM with disposals without an increase of the numerical effort. [Pineyro and Viera \(2010\)](#) also considered the possibility to dispose returned products. A tabu search heuristic was suggested to solve this kind of lot-sizing problem. [Li, Chen, and Cai \(2006\)](#) presented an uncapacitated multi-product lot-sizing problem. However, the interactions of these products by competing for the scarce capacity are neglected.

To the best of our knowledge, only a few papers addressed capacitated lot-sizing problems with product returns and remanufacturing. [Li, Chen, and Cai \(2007\)](#) described a model formulation for a lot-sizing problem with product substitution, i.e., the remanufactured product serves as a substitute for the serviceable. The production capacity is limited, while setup times are neglected. A genetic algorithm is applied to determine a produc-

Figure 1: Dynamic Capacitated Lot-Sizing with Product Returns and Remanufacturing (Schulz 2011: 2521)



tion plan. Pan, Tang, and Liu (2009) investigated a single-product capacitated lot-sizing problem. Here, the number of produced, remanufactured and disposed items are restricted in each period. Akcali and Cetinkaya (2011) pointed out that most of the lot-sizing model formulations with product returns and remanufacturing in the existing literature only investigate single-item problems. Therefore, the interactions between several items produced on the same production system are totally neglected. Neglecting these interactions often leads to an infeasible production plan if these items compete for the scarce capacities of the production system. Hence, we present in this paper a new model formulation for the multi-product capacitated lot-sizing problem with product returns and remanufacturing. This CLSP-RM is solved by combining a column-generation approach and a truncated branch-and-bound method.

3 Dynamic Capacitated Lot-Sizing with Product Returns and Remanufacturing

3.1 Model Formulation of the CLSP-RM

The objective of the capacitated lot-sizing problem with product returns and remanufacturing is to determine a production schedule for K products ($k = 1, \dots, K$) which minimizes the sum of setup, production, remanufacturing and holding costs over a planning horizon of T periods ($t = 1, \dots, T$).

This production schedule includes the production quantities Q_{kt} and the remanufacturing quantities Q_{kt}^r as well as the end-of-period inventory levels of the serviceables Y_{kt} and of the recoverables Y_{kt}^r of product k in period t .

For each product k the external demand d_{kt} has to be fulfilled in the respective period t , while backlogging is not allowed. Serviceables of product k can be held in stock (Y_{kt}) to satisfy the demand in later periods at holding costs hc_k per unit and period. The quantities of returned items r_{kt} of product k in period t are also known in advance. The returned items can be remanufactured (Q_{kt}^r) to become as good as new, i.e. serviceables. Furthermore, returned products can also be held in stock as recoverables (Y_{kt}^r) at holding costs hc_k^r per unit and period to be remanufactured in later periods. The variable production costs are pc_k and the remanufacturing costs are pc_k^r for one unit of product k .

The products are (re)manufactured on a production system, which consists of two separated resources: On the first resource, the products can only be manufactured, while remanufacturing is located on the second resource. For this reason, a (separate) setup is required for producing and another setup for remanufacturing. For manufacturing product k , a setup leads to setup costs sc_k and setup times ts_k . If product k is manufactured in period t , the setup variable γ_{kt} equals 1, otherwise $\gamma_{kt} = 0$. For remanufacturing product k , different setup costs sc_k^r and setup times ts_k^r are considered.

A further setup variable γ_{kt}^r is used to model a setup for the recoverables of product k in period t . Analogously, the setup variable γ_{kt}^r of product k in period t equals 1, if product k is remanufactured in the respective period, otherwise $\gamma_{kt}^r = 0$. The capacity c_t of the manufacturing system and the capacity c_t^r of the remanufacturing system are given in each period t . The production times are tp_k and the remanufacturing times are tp_k^r for product k . The described production process is visualized in Figure 1.

The CLSP-RM with separate setups (CLSP-RM-SS) can be stated using the notation in Table 1:

Model CLSP-RM-SS:

$$(1) \quad \min Z = \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} (hc_k \cdot Y_{kt} + hc_k^r \cdot Y_{kt}^r) \\ + \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} (pc_k \cdot Q_{kt} + pc_k^r \cdot Q_{kt}^r) \\ + \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} (sc_k \cdot \gamma_{kt} + sc_k^r \cdot \gamma_{kt}^r)$$

subject to

$$(2) \quad Y_{k,t-1} + Q_{kt} + Q_{kt}^r = d_{kt} + Y_{kt} \quad \forall k, t$$

$$(3) \quad Y_{k,t-1}^r + r_{kt} = Q_{kt}^r + Y_{kt}^r \quad \forall k, t$$

$$(4) \quad \sum_{k \in \mathcal{K}} (tp_k \cdot Q_{kt} + ts_k \cdot \gamma_{kt}) \leq c_t \quad \forall t$$

$$(5) \quad \sum_{k \in \mathcal{K}} (tp_k^r \cdot Q_{kt}^r + ts_k^r \cdot \gamma_{kt}^r) \leq c_t^r \quad \forall t$$

$$(6) \quad Q_{kt} \leq M_{kt} \cdot \gamma_{kt} \quad \forall k, t$$

$$(7) \quad Q_{kt}^r \leq M_{kt}^r \cdot \gamma_{kt}^r \quad \forall k, t$$

$$(8) \quad Q_{kt}, Q_{kt}^r, Y_{kt}, Y_{kt}^r \geq 0 \quad \forall k, t$$

$$(9) \quad \gamma_{kt}, \gamma_{kt}^r \in \{0, 1\} \quad \forall k, t$$

The objective function (1) minimizes the sum of inventory holding, production, remanufacturing and

Table 1: Notation used for the CLSP-RM

Indices and index sets:

\mathcal{K} set of products ($k \in \{1, \dots, K\}$)

\mathcal{T} set of periods ($t \in \{1, \dots, T\}$)

Parameters:

c_t available capacity of the resource for manufacturing in period t

c_t^r available capacity of the resource for remanufacturing in period t

d_{kt} external demand of product k in period t

hc_k holding cost of product k per unit and period

hc_k^r holding cost for returns of product k per unit and period

M_{kt} big number for product k in period t

pc_k production cost of product k per unit

pc_k^r remanufacturing cost of product k per unit

r_{kt} returns of product k in period t

sc_k setup cost of product k

sc_k^r setup cost of returned product k

tp_k production time for one unit of product k

tp_k^r remanufacturing time for one unit of product k

ts_k setup time of product k

ts_k^r setup time of returned product k

Decision variables:

Q_{kt} production quantity of product k in period t

Q_{kt}^r remanufacturing quantity of product k in period t

Y_{kt} inventory of product k at the end of period t

Y_{kt}^r inventory of recoverables of product k at the end of period t

γ_{kt} binary (joint) setup variable of product k in period t

γ_{kt}^r binary setup variable for remanufacturing product k in period t

setup costs. Equations (2) and (3) represent the inventory balance constraints for the serviceables, respectively the recoverables. Equations (2) ensure that the given demand d_{kt} for product k is satisfied completely in each period t . Analogously, restrictions (3) guarantee that the returned items are either held in stock or remanufactured in the respective period. Constraints (4) and (5) contain the capacity restrictions. In both cases, the production

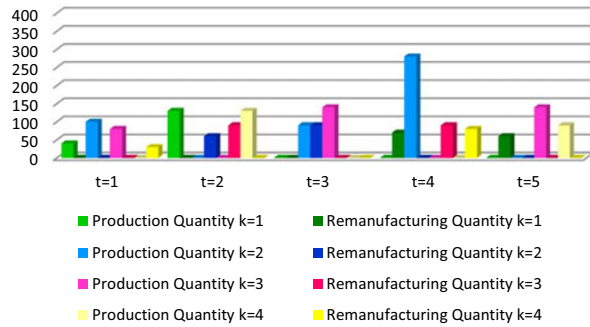
(re)manufacturing times and also the setup times must not exceed the given capacity. Constraints (6) and (7) link the production variables Q_{kt} and Q_{kt}^r to the respective binary setup variable γ_{kt} or γ_{kt}^r for product k in period t . Constraints (8) are the non-negativity constraints. The setup variables are defined to be binary, according to constraints (9). To tighten the linking constraints (6) and (7), the parameter M_{kt} can be determined as follows:

$$M_{kt} = \frac{\max\{c_t - ts_k, c_t^r - ts_k^r\}}{\min\{tp_k, tp_k^r\}} \quad \forall k, t$$

which describes the maximum (re)manufacturing quantity of product k that can possibly be (re)manufactured in period t due to the available capacity.

In the next step, we present a small numerical example for the CLSP-RM-SS with 4 products and 5 periods. The underlying parameter settings are given in Appendix 6. The optimal (re)manufacturing quantities are shown in Figure 2. A Gantt chart and the optimal production plan are given in Appendix 6 (Figure 7(a) and Table 10).

Figure 2: Numerical example for the CLSP-RM-SS with 4 products and 5 periods



The capacity of the resource for manufacturing is almost completely used. With the exception of period 3, products are either manufactured or remanufactured. Only one product is manufactured and remanufactured as well, e.g., product 2 in the third period. The total costs of the production plan amount to 9620.

In the case of joint setups, the products are (re)manufactured on the same resource with a given capacity c_t in each period t . A (joint) setup is necessary to prepare the production system to manufacture and/or remanufacture product k in

period t . This setup leads to setup costs sc_k and setup times ts_k . If product k is produced and/or remanufactured in period t , the respective setup variable γ_{kt} equals 1, otherwise $\gamma_{kt} = 0$. The CLSP-RM with joint setups (CLSP-RM-JS) can be modeled as follows.

Model CLSP-RM-JS:

$$(10) \quad \min Z = \sum_{k \in K} \sum_{t \in T} (hc_k \cdot Y_{kt} + hc_k^r \cdot Y_{kt}^r) + \sum_{k \in K} \sum_{t \in T} (pc_k \cdot Q_{kt} + pc_k^r \cdot Q_{kt}^r) + \sum_{k \in K} \sum_{t \in T} sc_k \cdot \gamma_{kt}$$

subject to

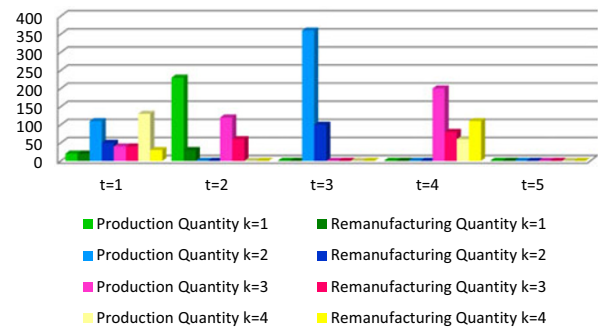
$$(2), (3), (8), (9)$$

$$(11) \quad \sum_{k \in K} (tp_k \cdot Q_{kt} + tp_k^r \cdot Q_{kt}^r + ts_k \cdot \gamma_{kt}) \leq c_t \quad \forall t$$

$$(12) \quad Q_{kt} + Q_{kt}^r \leq M_{kt} \cdot \gamma_{kt} \quad \forall k, t$$

In the objective function (10) of the CLSP-RM-JS, the setup costs sc_k^r can be omitted compared to (1). Due to the consideration of a common production system, only one capacity constraint (11) is required. Finally, the linking constraints (6) and (7) can be combined into constraints (12).

Figure 3: Numerical example for the CLSP-RM-JS with 4 products and 5 periods



Finally, the numerical example is also solved in the case of the CLSP-RM-JS. The optimal (re)manufacturing quantities are illustrated in Figure 3. A Gantt chart and the optimal production

plan are also provided in Appendix 6 (Figure 7(b) and Table 11).

In the case of the CLSP-RM-JS, the total costs amount to 6090 and are, therefore, lower compared to the CLSP-RM-SS. However, please note that the objective function values are not comparable as only one setup is required for (re)manufacturing in the case of the CLSP-RM-JS.

3.2 Complexity Results

Theorem 1 *The CLSP-RM is \mathcal{NP} -hard.*

Proof By setting $r_{kt} = 0$ for all products k and periods t the CLSP-RM can be reduced to the well-known CLSP. As [Florian, Lenstra, and Kan \(1980\)](#) showed that the CLSP is \mathcal{NP} -hard, the CLSP-RM is also \mathcal{NP} -hard. \square

Theorem 2 *For the CLSP-RM, the proof of the existence of a feasible solution is \mathcal{NP} -complete in the case of positive setup times ($ts_k > 0$).*

Proof As explained above, the CLSP-RM can be easily reduced to the CLSP. [Florian, Lenstra, and Kan \(1980\)](#) also proved that it is \mathcal{NP} -complete to prove the existence of a feasible solution for the CLSP in the case of positive setup times. Therefore, the same holds for the CLSP-RM. \square

3.3 Model Extensions of the CLSP-RM

Practical extensions of the CLSP-RM can be, e.g., the use of overtime or allowing back-orders. In each period the available capacity c_t of the resource for manufacturing can be extended by the use of overtime (O_t). The overtime costs are oc for each required unit of overtime. Analogously, the amount of overtime required on the resource for remanufacturing is denoted as O_t^r . Furthermore, in the case of a stock-out, the demand can be back-ordered; in this case a backlog B_{kt} has to be taken into account. The backlog costs are bc_k for product k .

In the following we present a model formulation for the CLSP-RM-SS with backlogs and overtime by using the additional notation presented in Table 2.

Model CLSP-RM-SS-BO-Ov:

$$(13) \quad \min Z = \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} (hc_k \cdot Y_{kt} + hc_k^r \cdot Y_{kt}^r)$$

$$\begin{aligned} &+ \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} (pc_k \cdot Q_{kt} + pc_k^r \cdot Q_{kt}^r) \\ &+ \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} (sc_k \cdot \gamma_{kt} + sc_k^r \cdot \gamma_{kt}^r) \\ &+ \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} bc_k \cdot B_{kt} + \sum_{t \in \mathcal{T}} oc \cdot (O_t + O_t^r) \end{aligned}$$

subject to

(3), (6) – (9)

$$(14) \quad Y_{k,t-1} + Q_{kt} + Q_{kt}^r + B_{kt} =$$

$$d_{kt} + Y_{kt} + B_{k,t-1} \quad \forall k, t$$

$$(15) \quad \sum_{k \in \mathcal{K}} (tp_k \cdot Q_{kt} + ts_k \cdot \gamma_{kt}) \leq c_t + O_t \quad \forall t$$

$$(16) \quad \sum_{k \in \mathcal{K}} (tp_k^r \cdot Q_{kt}^r + ts_k^r \cdot \gamma_{kt}^r) \leq c_t^r + O_t^r \quad \forall t$$

$$(17) \quad B_{kt}, O_t, O_t^r \geq 0 \quad \forall k, t$$

In the objective function (1) the backlog and overtime costs have to be taken into account, see (13). Furthermore, backlogs have to be considered in the inventory balance equation (14). The capacity restrictions (4) and (5) are extended by the use of overtime, see (15) and (16). Finally, backlogs and the amount of overtime are non-negative according to (17). The CLSP-RM-JS can be extended analogously.

It is worth mentioning that the CLSP-RM-SS-BO-Ov is no longer \mathcal{NP} -complete since backlogging and the use of overtime are not restricted.

Table 2: Additional notation used for the CLSP-RM with back-orders and overtime

Parameters:

bc_k backlog cost of product k per unit and period

oc overtime cost per unit

Decision variables:

B_{kt} backlog of product k in period t

O_t amount of overtime used by the resource for manufacturing in period t

O_t^r amount of overtime used by the resource for remanufacturing in period t

3.4 Valid Inequalities

As the well-known CLSP is a special case of the CLSP-RM, it is not surprising that first numerical experiments have shown that the LP relaxation of the proposed model formulations for the CLSP-RM yields very poor lower bounds. Thus, including valid inequalities in the model formulation is often promising for accelerating the solution process. Therefore, we use the adapted valid inequalities for the SLULSP-RM proposed by [Retel Helmrich, Jans, van den Heuvel, and Wagelmans \(2010\)](#).

In the case of separate setups, the valid inequalities are defined as follows:

$$(18) \quad Y_{kt} \geq \sum_{s=t+1}^{t+p} d_{ks} - \sum_{s=t+1}^{t+p} M_{ks} \cdot (y_{ks} + y_{ks}^r)$$

$$\forall k, t = 1, \dots, T-1, p = 1, \dots, T-t$$

$$(19) \quad Y_{kt}^r \geq \sum_{s=t-p}^t r_{ks} - \sum_{s=t-p}^t M_{ks} \cdot y_{ks}^r$$

$$\forall k, t = 2, \dots, T, p = 1, \dots, t-1$$

The valid inequalities (18) can be interpreted as follows: If we assume no setup activities for product k in the periods $t+1, \dots, t+p$, the end-of-period inventory Y_{kt} must be high enough to fulfill the total demand in the periods $t+1, \dots, t+p$. Otherwise, if the end-of-period inventory Y_{kt} is not sufficient at least one setup is required in the periods $t+1, \dots, t+p$. Analogously, inequalities (19) are defined for the recoverables.

If joint setups are considered, the valid inequalities can be defined as follows:

$$(20) \quad Y_{kt} \geq \sum_{s=t+1}^{t+p} d_{ks} - \sum_{s=t+1}^{t+p} M_{ks} \cdot y_{ks}$$

$$\forall k, t = 1, \dots, T-1, p = 1, \dots, T-t$$

$$(21) \quad Y_{kt}^r \geq \sum_{s=t-p}^t r_{ks} - \sum_{s=t-p}^t M_{ks} \cdot y_{ks}$$

$$\forall k, t = 2, \dots, T, p = 1, \dots, t-1$$

The interpretation of the valid inequalities (20) and (21) is similar to the explanation of (18) and

(19). Obviously, the valid inequalities are also valid for the CLSP-RM, as the demand has to be fulfilled without backlogging and disposals are not allowed. The adaptation of the VI in the case of back-orders can be found in Appendix 6.

Due to the \mathcal{NP} -hardness of the CLSP-RM it is nearly impossible to solve problem instances of practical size to optimality within a reasonable time. Therefore, we propose an MP-based solution approach to determine high-quality solutions. At first, a column-generation (CG) approach ([Haase 2005](#)) is applied to generate tight lower bounds for the CLSP-RM. Afterwards, a truncated branch-and-bound approach is used to construct a solution based on the lower bound provided by CG.

4 A Column-Generation Approach

4.1 Basic Idea

The general idea of a column-generation approach can be described as follows ([Bahl 1983](#); [Cattrysse, Maes, and van Wassenhove 1990](#)): A reformulation of the CLSP-RM is used as a so-called master problem. At first, the master problem includes only a very limited number of columns. Iteratively, new columns are generated for the master problem by solving a respective subproblem. If the new column promises a reduction of the objective function value, this new column is included in the master problem. New columns are generated iteratively as long as they reduce the current objective function's value of the master problem.

In the following, we only present the proceeding of the CG approach for the CLSP-RM with separate setups. Analogously, the CG approach can be adapted to the CLSP-RM with joint setups straightforwardly.

4.2 The Master Problem – A Set Partitioning Problem

For the CG approach, we use a reformulation of the CLSP-RM, namely the *Set Partitioning Problem* (SPP, [Garfinkel and Nemhauser 1969](#); [Cattrysse, Maes, and van Wassenhove 1990](#)) with capacity restrictions, as the LP relaxation of the SPP provides a substantially tighter lower bound ([Haase 2005](#)) than the LP relaxation of the presented model formulation.

The objective of the SPP is to choose exactly one production schedule for each product k at minimal

costs so that the capacity constraints are met in each period. Therefore, a set of production schedules S_k exists for each product k which has to be generated in advance. For each production schedule $n \in S_k$ of product k the total costs nc_{kn} (including setup, (re)manufacturing and holding costs) are known. Furthermore, the capacity consumptions tp_{ktn} (including manufacturing and setup times) and tp_{ktn}^r (including remanufacturing and setup times) can also be derived. For the choice of a schedule n for product k a binary variable δ_{kn} is used. Here, δ_{kn} equals 1, if schedule n is chosen for product k and otherwise $\delta_{kn} = 0$.

Table 3: Additional notation used for the SPP

<u>Indices and index sets:</u>	
S_k	set of schedules of product k ($n \in \{1, \dots, S_k\}$)
<u>Parameters:</u>	
nc_{kn}	cost of schedule n of product k
tp_{ktn}	capacity consumption of schedule n for manufacturing product k in period t
tp_{ktn}^r	capacity consumption of schedule n for remanufacturing product k in period t
<u>Decision variables:</u>	
δ_{kn}	binary variable for choosing schedule n of product k

In the following, we present the model formulation of the SPP using the additional notation given in Table 3.

Model SPP:

$$(22) \quad \min Z = \sum_{k \in \mathcal{K}} \sum_{n \in S_k} nc_{kn} \cdot \delta_{kn}$$

subject to

$$(23) \quad \sum_{n \in S_k} \delta_{kn} = 1 \quad \forall k$$

$$(24) \quad \sum_{k \in \mathcal{K}} \sum_{n \in S_k} tp_{ktn} \cdot \delta_{kn} \leq c_t \quad \forall t$$

$$(25) \quad \sum_{k \in \mathcal{K}} \sum_{n \in S_k} tp_{ktn}^r \cdot \delta_{kn} \leq c_t^r \quad \forall t$$

$$(26) \quad \delta_{kn} \in \{0, 1\} \quad \forall k, n \in S_k$$

The objective function (22) minimizes the total costs. Equation (23) guarantees that exactly one schedule is chosen for each product k . The capacity constraints (24) and (25) ensure that the capacity consumption of all products does not exceed the given capacity in period t . Constraints (26) define the variables δ_{kn} to be binary. It is worth mentioning that the optimal solution of the SPP is also the optimal solution of the CLSP-RM if *all* possible production schedules are known in advance.

As the generation of all production schedules is quite time consuming, new schedules are determined iteratively and included in the SPP afterwards. Therefore, in each iteration a subproblem is solved in order to generate a new schedule n for each product k . This new production schedule n is included in the SPP, i.e. in the set S_k , if it promises a reduction of the current objective function value, i.e., the reduced costs are negative. However, if the number of schedules increases, the solution time to solve the SPP to optimality also increases substantially. Hence, to reduce the numerical effort, we only solve the LP relaxation of the SPP and thus, the CG approach terminates with a lower bound for the CLSP-RM.

4.3 The Subproblem – An SLULSP-RM

As a subproblem in the CG approach, we solve an SLULSP-RM (Teunter, Bayindir, and van den Heuvel 2006) to optimality for each product k . Numerical investigations have shown that the numerical effort to solve an instance of the SLULSP-RM with separate (or joint) setups to optimality is almost negligible using CPLEX with the VIs (18) and (19), respectively (20) and (21). The SLULSP-RM _{k} can be stated for a respective product k using the dual variables σ_k , π_t , and π_t^r of the relaxed SPP. Here, σ_k denotes the dual variable which corresponds to equation (23) for product k . The dual variables (or shadow prices) π_t and π_t^r correspond to the capacity constraint (24), respectively (25) of period t .

Model SLULSP-RM _{k} :

$$(27) \quad \min Z_k = \sum_{t \in \mathcal{T}} (hc_k \cdot Y_{kt} + hc_k^r \cdot Y_{kt}^r) + \sum_{t \in \mathcal{T}} (pc_k \cdot Q_{kt} + pc_k^r \cdot Q_{kt}^r)$$

$$\begin{aligned}
& + \sum_{t \in \mathcal{T}} (sc_k \cdot \gamma_{kt} + sc_k^r \cdot \gamma_{kt}^r) - \sigma_k \\
& - \sum_{t \in \mathcal{T}} \pi_t \cdot (tp_k \cdot Q_{kt} + ts_k \cdot \gamma_{kt}) \\
& - \sum_{t \in \mathcal{T}} \pi_t^r \cdot (tp_k^r \cdot Q_{kt}^r + ts_k^r \cdot \gamma_{kt}^r)
\end{aligned}$$

subject to

(2), (3), (6)–(9)

The objective function (27) of the SLULSP-RM_k tends to minimize the reduced costs of the potentially new schedule for product k . The remaining constraints equal those of the CLSP-RM for the respective product k except for the capacity constraints (11). Each SLULSP-RM_k is solved to optimality using CPLEX. A new production schedule for product k is only included in the set S_k if the reduced costs of this new schedule are negative, i.e., the objective function value of the SLULSP-RM_k is negative.

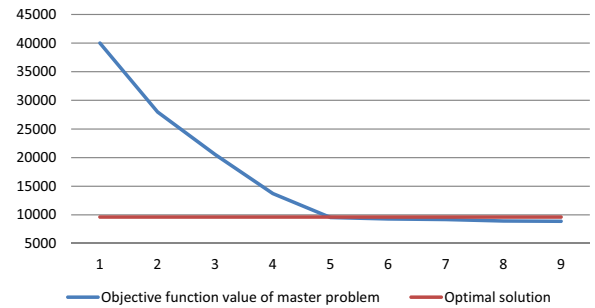
4.4 Outline of the Column-Generation Approach

To initialize the column-generation approach we start with a *dummy schedule* for each product k . Within a dummy schedule, no production and no setups take place, i.e., $q_{kt} = 0$ for all products k and all periods t . Hence, there exists no capacity consumption, i.e., $tp_{ktn} = tp_{ktn}^r = 0$. However, to avoid the choice of these schedules in the optimal solution, they are penalized at very high costs nc_{kn} . In each iteration, the LP relaxation of the master problem is solved first. Afterwards, the optimal solution of the related SLULSP-RM_k is determined for each product k by taking the dual variables σ_k , π_t , and π_t^r into account (see (27)). If the reduced costs of the corresponding new schedule, i.e., the objective function value of the related SLULSP-RM_k, are negative this new schedule is included in the set S_k . The algorithm terminates if the reduced costs are non-negative for all products in the current iteration. The LP relaxation of the SPP yields a lower bound for the CLSP-RM. Note that no feasible lower bound exists if any dummy schedule is selected in the solution of the LP relaxation after the column-generation approach terminates. In the next step, the lower bound obtained by the column-generation approach is used to generate

a feasible solution for the CLSP-RM by a truncated branch-and-bound approach using CPLEX. To minimize the numerical effort all binary setup variables are fixed for those products k with an integer solution in the LP relaxation of the SPP, i.e., $\exists n \in S_k$ with $\delta_{kn} = 1$. The respective setup variables γ_{kt} and γ_{kt}^r of product k are fixed related to the setup pattern, which corresponds to the production schedule n with $\delta_{kn} = 1$. After a given time limit the branch-and-bound approach (B&B) terminates with a production plan for the CLSP-RM. A flow chart of the CG approach is presented in Figure 5.

In Figure 4, the course of the objective function value of the relaxed SPP is displayed for the numerical example in section 3.1 in the case of the CLSP-RM-SS. After the generation of 9 production schedules for each product the CG approach terminates with a feasible lower bound.

Figure 4: Course of the objective function value of the master problem during the CG approach compared to the optimal solution



In Appendix 6 we present necessary adaptations of the solution approach in the case of back-orders and overtime.

5 Numerical Experiments

5.1 Description of the Test Design

We analyze the quality of our solution approach by defining five problem classes (PC) by varying the number of products and periods. Each problem class consists of 216 test instances (TI) leading to 1080 TI. Table 4 gives an overview of these five problem classes.

We vary different parameters to define the TI, e.g., the time between orders (TBO) to determine setup costs, the utilization *Util* of the production

Figure 5: Flow chart of Column-Generation Approach

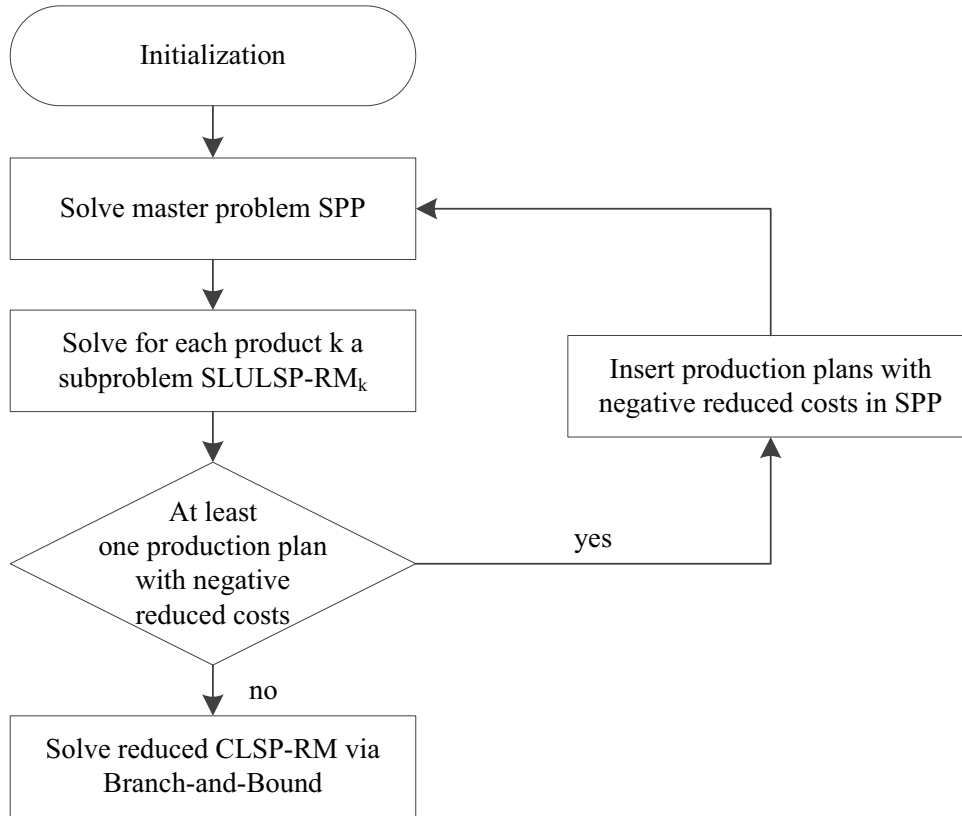


Table 4: Dimensions of the five problem classes

	K	T	#TI
Class 1	8	16	216
Class 2	10	24	216
Class 3	20	24	216
Class 4	40	24	216
Class 5	100	24	216

system, the consideration of setup times ts and the portion of holding costs hc^r for returned products compared to the holding costs of the serviceables. A detailed description of the TI including calculation rules is given in Appendix 6. Table 5 shows a list of these parameters and their range of values.

We implemented the model formulations and the solution approach in GAMS 23.7. For the solution of the subproblems and for the determination of reference values we used CPLEX 12.3. The MP-

Table 5: Varying parameters

TBO pattern	TBO	$\in \{1, 2, 4\}$
resource utilization	$Util$	$\in \{0.7, 0.8, 0.9\}$
setup time	ts	$\in \{0, 20\}$
holding costs of re-coverables	hc^r	$\in \{0.5, 0.7, 0.9\}$

based heuristic ran on an Intel Xeon CPU with a 2.93 GHz processor and 8 GB of RAM using two threads. The reference values are determined on the cluster TANE of the RRZN in Hannover using 4 parallel threads, each with a 2.93 GHz processor and 36 GB of RAM (<http://www.rrzn.uni-hannover.de/tane.html>).

5.2 Solution Quality of CPLEX Reference Values

The valid inequalities (18) and (19) are included in the CLSP-RM-SS to speed up the solution process

Table 6: Solution quality of CPLEX reference values

CLSP-RM-JS	TCPU ^{CPX}	TLim ^{CPX}	AvgGAP ^{CPX}	OptSol ^{CPX}
Class 1	37s	1h	0.00%	100.00%
Class 2	798s	2h	0.06%	93.06%
Class 3	3,085s	4h	0.03%	83.33%
Class 4	6,206s	8h	0.02%	79.63%
Class 5	13,712s	12h	0.02%	68.98%
CLSP-RM-SS	TCPU ^{CPX}	TLim ^{CPX}	AvgGAP ^{CPX}	OptSol ^{CPX}
Class 1	2,989s	1h	1.71%	22.69%
Class 2	7,180s	2h	3.21%	0.00%
Class 3	14,366s	4h	3.02%	0.00%
Class 4	28,803s	8h	3.37%	0.00%
Class 5	42,733s	12h	3.90%	0.00%

TCPU^{CPX} *Average solution time*
TLim^{CPX} *Given time limit*

AvgGAP^{CPX} *Average integrality gap*
OptSol^{CPX} *Portion of TI with a proven optimal solution*

of CPLEX. The valid inequalities (20) and (21) are used in the case of joint setups, respectively. In Table 6 the solution quality of the CPLEX reference values is reported, for both the case of joint and separate setups. The CPLEX reference values and the numerical results of the solution approach are provided for all TI online in the supplementary material.

A (feasible) solution has been found for all TI. The average solution time in seconds is given in “TCPU^{CPX}”. As we do not expect to solve all TI to optimality within a reasonable time, each TI is solved by a truncated branch-and-bound approach within a given time limit (column “TLim^{CPX}”). The average integrality gap is reported in column “AvgGAP^{CPX}”, where GAP_{TI}^{CPX} is derived as follows:

$$GAP_{TI}^{CPX} = \frac{(Sol_{TI}^{CPX} - LowB_{TI}^{CPX})}{Sol_{TI}^{CPX}} \cdot 100\%.$$

For each TI, Sol_{TI}^{CPX} describes the best objective function value found. $LowB_{TI}^{CPX}$ denotes the best lower bound obtained by CPLEX within the given time limit.

In the case of joint setups, all TI of PC 1 and a large number of TI of the remaining problem classes could be solved to optimality within the given time limit using the cluster TANE as reported in column “OptSol^{CPX}”. However, in the case of separate setups, CPLEX often fails to find the optimal solution within the given time limit due to the larger number of binary setup variables. In the following, the ref-

erence solutions obtained by CPLEX are denoted as CPLEX solutions or values.

5.3 Numerical Analysis of the Lower Bounds obtained by Column-Generation

To get an impression regarding the quality of the lower bound obtained by the column-generation approach, we first compare this lower bound with the CPLEX reference values. In Table 7, the average solution time in seconds to generate a feasible lower bound for the CLSP-RM is reported in column “TCPU^{CG}”. For each TI the deviation GAP_{TI}^{CG} of the CPLEX reference solution Sol_{TI}^{CPX} from the lower bound $LowB_{TI}^{CG}$ obtained by CG can be determined as follows:

$$GAP_{TI}^{CG} = \frac{(Sol_{TI}^{CPX} - LowB_{TI}^{CG})}{LowB_{TI}^{CG}} \cdot 100\%.$$

The average deviation is given in column “AvgGAP^{CG}”. Additionally, column “OptSol^{CG}” indicates the portion of TI whose lower bound equals the optimal solution. Finally, the portion of products with an integer solution in the lower bound is given in column “KFixed”. Table 7 gives an overview of the numerical results of the CG approach for all five problem classes.

For all TI a feasible lower bound is obtained by the column-generation approach. Here, a lower bound is called feasible if the LP relaxation does not include any dummy schedule. As expected,

Table 7: Numerical results of the lower bounds obtained by CG

CLSP-RM-JS	TCPU ^{CG}	AvgGAP ^{CG}	OptSol ^{CG}	KFixed
Class 1	2.4s	0.46%	25.46%	65.63%
Class 2	9.0s	0.26%	25.00%	66.44%
Class 3	16.0s	0.07%	27.78%	80.58%
Class 4	31.3s	0.03%	33.33%	91.28%
Class 5	82.9s	0.01%	35.65%	96.44%
CLSP-RM-SS	TCPU ^{CG}	AvgGAP ^{CG}	OptSol ^{CG}	KFixed
Class 1	15.5s	1.21%	5.09%	43.11%
Class 2	179.0s	1.14%	4.17%	39.72%
Class 3	299.9s	0.67%	6.48%	64.28%
Class 4	594.5s	0.64%	2.31%	82.70%
Class 5	1,574.3s	1.00%	0.00%	93.35%

TCPU^{CG} *Average solution time* *OptSol^{CG}* *Portion of optimally solved TI*
AvgGAP^{CG} *GAP of lower bound obtained by CG* *KFixed* *Portion of products with integer solution*

the run time increases with respect to the rising number of products.

In the case of joint setups, the average integrality GAP does not exceed 0.5%, while the average lower bound even amounts to only 0.01% in the case of the largest PC 5. Thus, the lower bound obtained by the CG approach is very close to the optimal solution. At least 25% of the TI could already be solved to optimality via CG. The portion of products with an integer solution ($\exists n \in \mathcal{S}_k$ with $\delta_{kn} = 1$) is quite high at more than 65%. The portion even rises to 96% for PC 5.

In the case of separate setups, the average solution time increases substantially due to the \mathcal{NP} -hardness of the SLULSP-RM with separate setups. The average integrality gap is almost comparable to the lower bound obtained for the case of joint setups. The portion of products with an integer solution ($\delta_{kn} = 1$) is only slightly lower compared to the portion of products in the case of joint setups. A detailed evaluation of our numerical results shows that an increase of setup costs and, therefore, of the TBO leads to a significant increase of the solution time. The integrality GAP also deteriorates with respect to the increase in setup costs. It is worth mentioning that at least 75% of the TI with low setup costs ($TBO = 1$) could even be solved to optimality with the CG approach in the case of joint setups. In summary, the solution quality decreases slightly if the setup cost rises. The numerical results of the CG approach with respect to

the varying setup costs can be found in Table 12.

In the case of scarce capacities, i.e., a higher utilization, the numerical effort rises as the solution times in column “TCPU^{CG}” shows. Simultaneously, the number of iterations increases as well. The average lower bound also rises in the case of a higher utilization. The numerical results with respect to the varying utilization of the production system are presented in Table 13. We observed that the impact of the remaining parameters on the solution quality is rather limited.

5.4 Numerical Analysis of the Upper Bounds obtained by the truncated Branch-and-Bound Method

In the next step, we investigate the solution quality of the production plans obtained by the truncated B&B method. Although a setup decision has already been fixed for a large number of products in the lower bound obtained by the CG approach, we do not expect to solve the remaining problem to optimality within a reasonable time. Therefore, a time limit is used for the truncated B&B approach. In Table 8 the numerical results of the upper bound obtained by the B&B approach are reported. In contrast to Table 7, the column “TCPU^{CG}” is substituted by “TCPU^{B&B}”. Here, the entries in column “TCPU^{B&B}” show the run time in seconds of the truncated B&B approach. For each TI the deviation $Dev_{TI}^{B\&B}$ of the CPLEX reference solution Sol_{TI}^{CPX} from the solution $Sol_{TI}^{B\&B}$ obtained by the

Table 8: Numerical results of the upper bounds obtained by the truncated B&B

CLSP-RM-JS	$TCPU^{B\&B}$	$AvgDev^{B\&B}$	$AvgGAP^{B\&B}$	$OptSol^{B\&B}$	$TLim^{B\&B}$
Class 1	4.8s	0.12%	0.58%	55.09%	50s
Class 2	19.9s	0.10%	0.37%	42.13%	100s
Class 3	26.9s	0.06%	0.13%	41.20%	150s
Class 4	34.8s	0.03%	0.05%	39.35%	300s
Class 5	59.8s	0.01%	0.02%	41.20%	600s
CLSP-RM-SS	$TCPU^{B\&B}$	$AvgDev^{B\&B}$	$AvgGAP^{B\&B}$	$BetSol^{B\&B}$	$TLim^{B\&B}$
Class 1	34.3s	0.66%	1.88%	30.09%	50s
Class 2	87.1s	1.60%	2.79%	18.06%	100s
Class 3	122.7s	1.78%	2.49%	43.52%	150s
Class 4	207.1s	0.29%	0.94%	78.70%	300s
Class 5	379.3s	-0.80%	0.19%	100.00%	600s

$TCPU^{B\&B}$	Solution time of truncated B&B	$OptSol^{B\&B}$	Portion of optimally solved TI
$AvgDev^{B\&B}$	Average deviation from CPLEX solution	$TLim^{B\&B}$	Given time limit
$AvgGAP^{B\&B}$	Average deviation from CG lower bound	$BetSol^{B\&B}$	Portion of better solved TI

truncated B&B approach can be determined as follows:

$$Dev_{TI}^{B\&B} = \frac{(Sol_{TI}^{CPX} - Sol_{TI}^{B\&B})}{Sol_{TI}^{CPX}} \cdot 100\%.$$

In column “ $AvgDev^{B\&B}$ ” the average deviation from the CPLEX reference solution is reported. Furthermore, we report the average gap based on the lower bound obtained by the CG approach in column “ $AvgGAP^{B\&B}$ ”, where $GAP^{B\&B}$ is defined as follows:

$$GAP_{TI}^{B\&B} = \frac{(Sol_{TI}^{B\&B} - LowB_{TI}^{CG})}{LowB_{TI}^{CG}} \cdot 100\%.$$

In the case of joint setups, the portion of optimally solved TI are given in “ $OptSol^{B\&B}$ ”. As the number of optimally solved TI is rather small in the case of separate setups, the column “ $OptSol^{B\&B}$ ” is substituted by the portion of TI which are solved at least as well as by CPLEX or even better. Therefore, the column is named “ $BetSol^{B\&B}$ ”. The given time limit is reported in “ $TLim^{B\&B}$ ”.

Due to the very small number of remaining *free* binary variables the numerical effort is rather limited. In the case of joint setups, the mean deviation from the CPLEX reference values ranges between 0.01% and 0.58%. This can be explained by the large portion of already fixed products after the CG approach (Table 7). More than 39% of all TI could be solved to optimality.

In the case of separate setups, the average deviation from the CPLEX reference values is also very moderate. For more than 18% of the TI the truncated B&B method finds a production plan which is at least as good as the CPLEX solution or even better. Furthermore, the truncated B&B determines for all TI a better solution than CPLEX in the case of PC 5. The low average GAP compared to the lower bound obtained by CG also demonstrates the high solution quality of the combined solution approach.

The numerical results of the truncated B&B method based on the different TBO patterns are reported in Table 14. Analogous to the observations regarding the CG approach, the solution time rises with respect to increasing setup costs. The same holds true for the solution quality. In the case of low setup costs ($TBO = 1$) the average deviation from the CPLEX reference values is lower compared to the average deviation in the case of high setup costs ($TBO = 4$). For low setup costs, at least 87.50% of all TI could be solved to optimality. This number decreases substantially to 0% with increasing setup costs. Furthermore, the total solution time of both approaches also grows according to rising setup costs.

For the sake of completeness, we give the numerical results of the truncated B&B method with respect to the utilization of the production system in Ta-

ble 15. The rise of solution times is also noticeable in the case of a higher utilization, while the average deviation from the CPLEX reference values deteriorates only marginally. Worth mentioning are the results of PC 5: Contrary to the results of the other classes, the solution quality does not decline in the cases of higher setup costs and/or higher utilization. This is due to the fact that CPLEX fails to find better solutions within the given time limit. We also investigated the solution quality of the proposed solution approach by allowing overtime and back-orders. The solution quality and the solution time are nearly comparable. Therefore, we abstain from showing these numerical results.

6 Summary and Outlook

In this paper, we presented a new model formulation for a capacitated lot-sizing problem with product returns and remanufacturing. In literature, only one-product problems have been considered so that the influence of different products on the same production systems are completely ignored. Furthermore, most of these approaches in literature neglect capacity restrictions. Thus, these production plans are of limited value as the capacity restrictions are often violated. Obviously, a rescheduling is necessary in these cases.

The proposed solution approach yields a very high solution quality. The lower bounds found by the CG approach are very close to the optimal solution as our numerical investigation shows. The combination of CG and a truncated branch-and-bound method leads to a very high solution quality and the solutions are very close to the CPLEX reference values, but with substantially reduced numerical effort. Furthermore, we have shown that an integrated (re)manufacturing planning can yield a cost reduction.

Future research should address the adaptation of further solution approaches, e.g., the Fix-and-Optimize heuristic by [Helber and Sahling \(2010\)](#). Possible extensions of the CLSP-RM will be linked lot sizes where the setup state of one product can be carried over to a subsequent period. Furthermore, different quality levels of the returned products should be taken into account. Different quality levels will result in varying remanufacturing costs and times.

Appendices

Appendix A: Description of the Numerical Example

For the numerical example in section 3.1, the setup costs sc_k and sc_k^r equal 500 for each product k . The holding costs are $hc_k = 1$ and $hc_k^r = 0.5$. The (re)manufacturing times tp_k and tp_k^r are assumed to be 1 and the setup times ts_k and ts_k^r are equal to 20 for each product k . The demand and returns are given in Table 9.

Table 9: Demand and return data of the numerical example in section 3.1

$k \setminus t$	d_{k1}	d_{k2}	d_{k3}	d_{k4}	d_{k5}
1	40	70	60	70	60
2	100	60	180	170	110
3	80	90	90	140	140
4	30	80	50	80	90

$k \setminus t$	r_{k1}	r_{k2}	r_{k3}	r_{k4}	r_{k5}
1	20	30	30	20	30
2	50	40	60	40	40
3	40	60	50	30	40
4	30	30	30	50	20

In the case of the CLSP-RM-SS, the capacities c_t and c_t^r amount to 300 capacity units in each period. In Table 10, the optimal production plan is given for the CLSP-RM-SS.

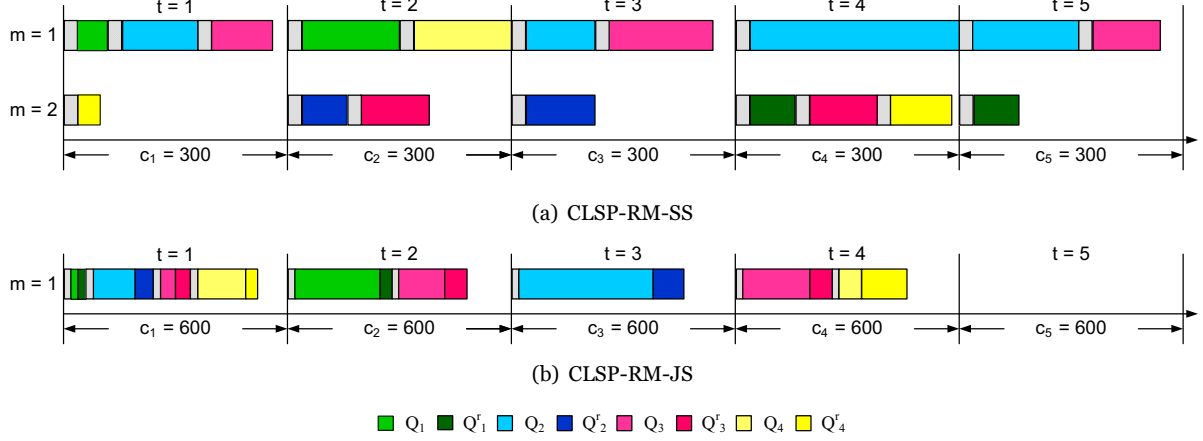
Table 10: Optimal production plan of the numerical example for CLSP-RM-SS

$k \setminus t$	q_{k1}	q_{k2}	q_{k3}	q_{k4}	q_{k5}
1	40	130	0	0	0
2	100	0	90	280	0
3	80	0	140	0	140
4	0	130	0	0	90

$k \setminus t$	q_{k1}^r	q_{k2}^r	q_{k3}^r	q_{k4}^r	q_{k5}^r
1	0	0	0	70	60
2	0	60	90	0	0
3	0	90	0	90	0
4	30	0	0	80	0

In the case of the CLSP-RM-JS, the capacity c_t amounts to 600 capacity units in each period. In Table 11, the optimal production plan is given for

Figure 6: Gantt charts of the numerical example with 4 products and 5 periods in section 3.1. Note that the sequences of the products presented in the Gantt charts are chosen arbitrarily since the sequences are not determined by the CLSP-RM.



the CLSP-RM-JS.

Table 11: Optimal production plan of the numerical example for CLSP-RM-JS

$k \setminus t$	q_{k1}	q_{k2}	q_{k3}	q_{k4}	q_{k5}
1	20	230	0	0	0
2	110	0	360	0	0
3	40	120	0	200	0
4	130	0	0	60	0

$k \setminus t$	q_{k1}^r	q_{k2}^r	q_{k3}^r	q_{k4}^r	q_{k5}^r
1	20	30	0	0	0
2	50	0	100	0	0
3	40	60	0	80	0
4	30	0	0	110	0

Appendix B: Adaptations in the Case of Back-Orders and Overtime

The valid inequalities (18) for the CLSP-RM-SS have to be adapted by taking backlogs into consideration.

$$(28) \quad Y_{kt} - B_{kt} \geq \sum_{s=t+1}^{t+p} d_{ks} - B_{k,t+p} - \sum_{s=t+1}^{t+p} M_{ks} \cdot (Y_{ks} + Y_{ks}^r) \quad \forall k, t = 1, \dots, T-1, p = 1, \dots, T-t$$

Analogously, the VI (20) for the CLSP-RM-JS must be changed.

For the solution approach, the use of overtime has to be considered only in the master problem. Therefore, the objective function (22), the capacity constraints (24) and (25) have to be extended.

Model SPP-Ov:

$$(29) \quad \min Z = \sum_{k \in \mathcal{K}} \sum_{n \in \mathcal{S}_k} nc_{kn} \cdot \delta_{kn} + \sum_{t \in \mathcal{T}} oc_t \cdot (O_t + O_t^r)$$

subject to

$$(23), (26)$$

$$(30) \quad \sum_{k \in \mathcal{K}} \sum_{n \in \mathcal{S}_k} tp_{ktn} \cdot \delta_{kn} \leq c_t + O_t \quad \forall t$$

$$(31) \quad \sum_{k \in \mathcal{K}} \sum_{n \in \mathcal{S}_k} tp_{ktn}^r \cdot \delta_{kn} \leq c_t^r + O_t^r \quad \forall t$$

Backlogs have to be considered only in the subproblem.

Model SLULSP-RM-BO_k:

$$(32) \quad \min Z_k = \sum_{t \in \mathcal{T}} (hc_k \cdot Y_{kt} + hc_k^r \cdot Y_{kt}^r) + \sum_{t \in \mathcal{T}} (pc_k \cdot Q_{kt} + pc_k^r \cdot Q_{kt}^r) + \sum_{t \in \mathcal{T}} (bc_k \cdot B_{kt})$$

$$- \sum_{t \in T} \pi_t \cdot (tp_k \cdot Q_{kt} + ts_k \cdot \gamma_{kt})$$

$$- \sum_{t \in T} \pi_t^r \cdot (tp_k^r \cdot Q_{kt}^r + ts_k^r \cdot \gamma_{kt}^r)$$

subject to

(3), (6)–(9), (14), (17)

However, further adaptations of the solution approach are not necessary.

Appendix C: Description of the Test Instances

At first, for each product k the average demand \bar{d}_k is generated randomly by following a uniform distribution in the interval $[50, 150]$. Based on the average demand \bar{d}_k , two demand series are provided based on a normal distribution with a different coefficient of variation for each series. Analogously, two return series for each product k are generated. However, the average return \bar{r}_k equals one-third of the average demand \bar{d}_k of product k . The demand and return series are provided online in the supplementary material.

The holding costs hc_k of the serviceables, the processing times tp_k and the remanufacturing times tp_k^r are assumed to be 1 for each product k . The holding costs hc_k^r of the recoverables are assumed to be lower than the holding costs hc_k of the serviceables. For the holding costs hc_k^r three scenarios are defined (Table 5). The production and remanufacturing costs pc_k and pc_k^r are assumed to be equal and therefore not crucial.

The setup costs sc_k are determined based on the average demand \bar{d}_k of product k and the respective time between orders (TBO):

$$sc_k = \frac{\bar{d}_k \cdot TBO^2 \cdot hc_k}{2} \quad \forall k.$$

In the case of separate setups, the setup costs sc_k^r for remanufacturing equals the setup costs sc_k .

The setups times ts_k (and ts_k^r) are assumed to be zero or equal to 20 for all products. To derive the capacity c_t in the case of joint setups, we assume a setup activity in each period t for all products and allow the production of the average demand \bar{d}_k ; this leads to

$$c_t = \sum_k \left(\max\{tp_k, tp_k^r\} \cdot \bar{d}_k + ts_k \right) \quad \forall t.$$

In the case of separate setups, we also assume a setup activity on both production systems in each period t for all products. However, the average demand \bar{d}_k is reduced by the average return \bar{r}_k . This leads to

$$c_t = \sum_k \left(tp_k \cdot (\bar{d}_k - \bar{r}_k) + ts_k \right) \quad \forall t.$$

Additionally, the capacity c_t^r is defined according to the average return \bar{r}_k and allowing also a setup activity in each period:

$$c_t^r = \sum_k \left(tp_k^r \cdot \bar{r}_k + ts_k^r \right) \quad \forall t.$$

To guarantee a large amount of feasible solutions, the available capacity is then extended with respect to the given utilization $Util$

$$c_t = \frac{c_t}{Util} \quad \forall t.$$

The capacity c_t^r of the remanufacturing system is extended analogously.

Table 12: Numerical results of CG approach related to TBOs

CLSP-RM-JS	TCPU ^{CG}	AvgGAP ^{CG}	OptSol ^{CG}	KFixed
Class 1				
TBO = 1	0.6s	0.10%	75.00%	90.45%
TBO = 2	1.5s	0.20%	1.39%	67.01%
TBO = 4	5.0s	1.08%	0.00%	39.41%
Class 2				
TBO = 1	1.1s	0.03%	75.00%	95.83%
TBO = 2	4.2s	0.11%	0.00%	64.44%
TBO = 4	21.7s	0.64%	0.00%	39.03%
Class 3				
TBO = 1	2.1s	0.01%	83.33%	98.33%
TBO = 2	7.9s	0.03%	0.00%	80.28%
TBO = 4	38.0s	0.17%	0.00%	63.13%
Class 4				
TBO = 1	3.2s	0.00%	95.83%	99.90%
TBO = 2	15.6s	0.01%	4.17%	91.60%
TBO = 4	75.1s	0.07%	0.00%	82.36%
Class 5				
TBO = 1	7.6s	0.00%	100.00%	100.00%
TBO = 2	39.5s	0.00%	6.94%	96.64%
TBO = 4	201.6s	0.03%	0.00%	92.68%
CLSP-RM-SS	TCPU ^{CG}	AvgGAP ^{CG}	OptSol ^{CG}	KFixed
Class 1				
TBO = 1	5.5s	0.24%	15.28%	65.10%
TBO = 2	14.9s	0.76%	0.00%	41.84%
TBO = 4	26.1s	2.63%	0.00%	22.40%
Class 2				
TBO = 1	48.8s	0.17%	12.50%	59.03%
TBO = 2	196.7s	0.72%	0.00%	39.44%
TBO = 4	291.5s	2.54%	0.00%	20.69%
Class 3				
TBO = 1	62.8s	0.06%	19.44%	81.67%
TBO = 2	279.8s	0.43%	0.00%	64.65%
TBO = 4	557.1s	1.53%	0.00%	46.53%
Class 4				
TBO = 1	94.9s	0.03%	6.94%	93.96%
TBO = 2	541.7s	0.38%	0.00%	84.55%
TBO = 4	1,146.7s	1.51%	0.00%	69.58%
Class 5				
TBO = 1	187.7s	0.06%	0.00%	98.71%
TBO = 2	1,614.9s	0.78%	0.00%	94.17%
TBO = 4	2,920.1s	2.16%	0.00%	87.17%

Table 13: Numerical results of CG approach related to the resource utilization

CLSP-RM-JS	TCPU ^{CG}	AvgGAP ^{CG}	OptSol ^{CG}	KFixed
Class 1				
<i>Util</i> = 0.7	1.7s	0.16%	34.72%	77.08%
<i>Util</i> = 0.8	2.2s	0.32%	33.33%	68.58%
<i>Util</i> = 0.9	3.2s	0.91%	8.33%	51.22%
Class 2				
<i>Util</i> = 0.7	6.4s	0.08%	33.33%	75.69%
<i>Util</i> = 0.8	8.9s	0.18%	33.33%	68.47%
<i>Util</i> = 0.9	11.7s	0.52%	8.33%	55.14%
Class 3				
<i>Util</i> = 0.7	13.8s	0.02%	33.33%	88.33%
<i>Util</i> = 0.8	15.5s	0.05%	33.33%	81.53%
<i>Util</i> = 0.9	18.7s	0.14%	16.67%	71.88%
Class 4				
<i>Util</i> = 0.7	24.1s	0.01%	37.50%	94.65%
<i>Util</i> = 0.8	31.3s	0.02%	33.33%	91.42%
<i>Util</i> = 0.9	38.5s	0.05%	29.17%	87.78%
Class 5				
<i>Util</i> = 0.7	63.2s	0.00%	40.28%	97.90%
<i>Util</i> = 0.8	88.7s	0.01%	33.33%	96.56%
<i>Util</i> = 0.9	96.8s	0.02%	33.33%	94.86%
CLSP-RM-SS	TCPU ^{CG}	AvgGAP ^{CG}	OptSol ^{CG}	KFixed
Class 1				
<i>Util</i> = 0.7	10.9s	0.61%	13.89%	57.81%
<i>Util</i> = 0.8	15.0s	1.08%	1.39%	41.84%
<i>Util</i> = 0.9	20.6s	1.93%	0.00%	29.69%
Class 2				
<i>Util</i> = 0.7	134.4s	0.53%	12.50%	53.33%
<i>Util</i> = 0.8	180.9s	1.03%	0.00%	42.08%
<i>Util</i> = 0.9	221.7s	1.87%	0.00%	23.75%
Class 3				
<i>Util</i> = 0.7	246.1s	0.34%	13.89%	76.88%
<i>Util</i> = 0.8	292.6s	0.64%	5.56%	65.49%
<i>Util</i> = 0.9	361.0s	1.03%	0.00%	50.49%
Class 4				
<i>Util</i> = 0.7	547.1s	0.38%	4.17%	90.07%
<i>Util</i> = 0.8	615.5s	0.59%	2.78%	84.51%
<i>Util</i> = 0.9	620.9s	0.95%	0.00%	73.51%
Class 5				
<i>Util</i> = 0.7	1,442.6s	0.68%	0.00%	96.33%
<i>Util</i> = 0.8	1,559.7s	0.93%	0.00%	94.15%
<i>Util</i> = 0.9	1,720.5s	1.39%	0.00%	89.56%

Table 14: Numerical results of the upper bounds obtained by the truncated B&B related to TBOs

CLSP-RM-JS	TCPU ^{B&B}	AvgDev ^{B&B}	AvgGAP ^{B&B}	OptSol ^{B&B}
Class 1				
TBO = 1	0.1s	0.00%	0.11%	97.22%
TBO = 2	0.3s	0.05%	0.25%	50.00%
TBO = 4	14.1s	0.31%	1.39%	18.06%
Class 2				
TBO = 1	0.1s	0.00%	0.03%	97.22%
TBO = 2	1.7s	0.05%	0.16%	25.00%
TBO = 4	58.0s	0.27%	0.91%	4.17%
Class 3				
TBO = 1	0.1s	0.00%	0.01%	87.50%
TBO = 2	1.2s	0.01%	0.04%	33.33%
TBO = 4	79.4s	0.18%	0.35%	2.78%
Class 4				
TBO = 1	0.2s	0.00%	0.00%	100.00%
TBO = 2	0.7s	0.01%	0.01%	18.06%
TBO = 4	103.5s	0.08%	0.15%	0.00%
Class 5				
TBO = 1	0.4s	0.00%	0.00%	100.00%
TBO = 2	1.3s	0.00%	0.01%	18.06%
TBO = 4	177.8s	0.02%	0.05%	5.56%
CLSP-RM-SS	TCPU ^{B&B}	AvgDev ^{B&B}	AvgGAP ^{B&B}	BetSol ^{B&B}
Class 1				
TBO = 1	14.3s	0.08%	0.32%	56.94%
TBO = 2	39.1s	0.42%	1.19%	27.78%
TBO = 4	49.7s	1.46%	4.14%	5.56%
Class 2				
TBO = 1	65.5s	0.11%	0.27%	34.72%
TBO = 2	95.6s	0.92%	1.66%	19.44%
TBO = 4	100.0s	3.78%	6.45%	0.00%
Class 3				
TBO = 1	77.9s	0.01%	0.07%	51.39%
TBO = 2	140.1s	0.46%	0.90%	63.89%
TBO = 4	150.1s	4.86%	6.49%	15.28%
Class 4				
TBO = 1	88.3s	-0.02%	0.01%	87.50%
TBO = 2	233.0s	-0.15%	0.23%	86.11%
TBO = 4	300.1s	1.04%	2.58%	62.50%
Class 5				
TBO = 1	98.0s	-0.06%	0.00%	100.00%
TBO = 2	439.8s	-0.71%	0.06%	100.00%
TBO = 4	600.2s	-1.62%	0.50%	100.00%

Table 15: Numerical results of the upper bounds obtained by the truncated B&B related to the resource utilization

CLSP-RM-JS	TCPU ^{B&B}	AvgDev ^{B&B}	AvgGAP ^{B&B}	OptSol ^{B&B}
Class 1				
Util = 0.7	0.6s	0.09%	0.26%	65.28%
Util = 0.8	3.9s	0.11%	0.43%	55.56%
Util = 0.9	10.0s	0.15%	1.06%	44.44%
Class 2				
Util = 0.7	4.7s	0.03%	0.11%	47.22%
Util = 0.8	20.9s	0.07%	0.25%	47.22%
Util = 0.9	34.2s	0.21%	0.74%	31.94%
Class 3				
Util = 0.7	8.2s	0.02%	0.04%	50.00%
Util = 0.8	25.7s	0.04%	0.09%	45.83%
Util = 0.9	46.7s	0.13%	0.27%	27.78%
Class 4				
Util = 0.7	4.4s	0.01%	0.02%	47.22%
Util = 0.8	27.5s	0.02%	0.03%	37.50%
Util = 0.9	72.6s	0.05%	0.11%	33.33%
Class 5				
Util = 0.7	4.9s	0.00%	0.01%	51.39%
Util = 0.8	49.7s	0.01%	0.01%	37.50%
Util = 0.9	124.9s	0.01%	0.03%	34.72%
CLSP-RM-SS	TCPU ^{B&B}	AvgDev ^{B&B}	AvgGAP ^{B&B}	BetSol ^{B&B}
Class 1				
Util = 0.7	25.2s	0.24%	0.85%	43.06%
Util = 0.8	35.1s	0.64%	1.74%	23.61%
Util = 0.9	42.7s	1.09%	3.06%	23.61%
Class 2				
Util = 0.7	76.2s	0.93%	1.47%	30.56%
Util = 0.8	85.9s	1.54%	2.62%	15.28%
Util = 0.9	99.0s	2.34%	4.29%	8.33%
Class 3				
Util = 0.7	99.4s	0.52%	0.87%	61.11%
Util = 0.8	124.8s	1.71%	2.38%	41.67%
Util = 0.9	143.9s	3.10%	4.21%	27.78%
Class 4				
Util = 0.7	155.1s	-0.20%	0.18%	94.44%
Util = 0.8	215.7s	0.16%	0.75%	84.72%
Util = 0.9	250.7s	0.91%	1.88%	56.94%
Class 5				
Util = 0.7	286.9s	-0.63%	0.05%	100.00%
Util = 0.8	353.6s	-0.78%	0.14%	100.00%
Util = 0.9	497.5s	-0.99%	0.37%	100.00%

Acknowledgements

I would like to thank both referees and the department editor for their very helpful comments which helped me to improve this paper, and Paul Cochrane for his valuable support during the numerical study.

References

- Akcali, Emel and Sila Cetinkaya (2011): Quantitative Models for Inventory and Production Planning in Closed-Loop Supply Chains, *International Journal of Production Research*, 49 (8): 2373-2407.
- Bahl, Harish C. (1983): Column Generation Based Heuristic Algorithm for Multi-Item Scheduling, *IIE Transactions*, 15 (2): 136-141.
- Barany, Imre, Tony J. van Roy, and Laurence A. Wolsey (1984): Strong Formulations for Multi-Item Capacitated Lot Sizing, *Management Science*, 30 (10): 1255-1261.
- Buschkühl, Lisbeth, Florian Sahling, Stefan Helber, and Horst Tempelmeier (2010): Dynamic Capacitated Lot-Sizing Problems: A Classification and Review of Solution Approaches, *OR Spectrum*, 32 (2): 231-261.
- Cattrysse, Dirk, Johan Maes, and Luk N. van Wassenhove (1990): Set Partitioning and Column Generation Heuristics for Capacitated Dynamic Lot Sizing, *European Journal of Operational Research*, 46 (1): 38-47.
- Fleischmann, Moritz, Jacqueline M. Bloemhof-Ruwaard, Rommert Dekker, Erwin van der Laan, Jo A. E. E. van Nunen, and Luk N. van Wassenhove (1997): Quantitative Models for Reverse Logistics: A review, *European Journal of Operational Research*, 103 (1): 1-17.
- Florian, Michael, Jan Karel Lenstra, and Alexander R. Kan (1980): Deterministic Production Planning: Algorithms and Complexity, *Management Science*, 26 (7): 669-679.
- Garfinkel, Robert S. and George L. Nemhauser (1969): Set-Partitioning Problem: Set Covering With Equality Constraints, *Operations Research*, 17 (5): 848-856.
- Guide, V. Daniel R. and Luk N. van Wassenhove (2001): Managing Product Returns for Remanufacturing, *Production and Operations Management*, 10 (2): 142-155.
- Haase, Knut (2005): Solving Large-Scale Capacitated Lot-Sizing Problems Close to Optimality, Working Paper, Technische Universität Dresden.
- Helber, Stefan and Florian Sahling (2010): A Fix-and-Optimize Approach for the Multi-Level Capacitated Lot Sizing Problem, *International Journal of Production Economics*, 123 (2): 247-256.
- Karimi, B., Seyyed M. T. Fatemi Ghomi, and J. M. Wilson (2003): The Capacitated Lot Sizing Problem: A Review of Models and Algorithms, *Omega*, 31 (5): 365-378.
- Li, Yongjian, Jian Chen, and Xiaoqiang Cai (2006): Uncapacitated Production Planning With Multiple Product Types, Returned Product Remanufacturing, and Demand Substitution, *OR Spectrum*, 28 (1): 101-125.
- Li, Yongjian, Jian Chen, and Xiaoqiang Cai (2007): Heuristic Genetic Algorithm for Capacitated Production Planning Problems With Batch Processing and Remanufacturing, *International Journal of Production Economics*, 105 (2): 301-317.
- Pan, Zhendong, Jiafu Tang, and Ou Liu (2009): Capacitated Dynamic Lot Sizing Problems in Closed-Loop Supply Chain, *European Journal of Operational Research*, 198 (3): 810-821.
- Pineyro, Pedro and Omar Viera (2010): The Economic Lot-Sizing Problem With Remanufacturing and One-Way Substitution, *International Journal of Production Economics*, 124 (2): 482-488.
- Quadt, Daniel and Heinrich Kuhn (2008): Capacitated Lot-Sizing With Extensions: A Review, *4OR*, 6 (1): 61-83.
- Quariguasi Frota Neto, Joao, G. Walther, Jacqueline Bloemhof, J. A. E. E. van Nunen, and T. Spengler (2009): A Methodology for Assessing Eco-Efficiency in Logistics Networks, *European Journal of Operational Research*, 193 (3): 670-682.
- Retel Helmrich, Mathijn J., Raf Jans, Wilco van den Heuvel, and Albert P. M. Wagelmans (2010): Economic Lot-Sizing With Remanufacturing: Complexity and Efficient Formulations, <http://people.few.eur.nl/wvandenheuvel/papers/ei201071.pdf> (Access date: 2011-09-12).
- Richter, Knut and Mirko Sombrutzki (2000): Remanufacturing Planning for the Reverse Wagner/Whitin Models, *European Journal of Operational Research*, 121 (2): 304-315.
- Richter, Knut and Jens Weber (2001): The Reverse Wagner/Whitin Model With Variable Manufacturing and Remanufacturing Cost, *International Journal of Production Economics*, 71 (1-3): 447-456.
- Rubio, Sergio, Antonio Chamorro, and Francisco J. Miranda (2008): Characteristics of the Research on Reverse Logistics (1995-2005), *International Journal of Production Research*, 46 (4): 1099-1120.
- Sbihi, Abdelkader and Richard W. Eglese (2010): Combinatorial Optimization and Green Logistics, *Annals of Operations Research*, 175 (1): 159-175.
- Schulz, Tobias (2011): A New Silver-Meal Based Heuristic for the Single-Item Dynamic Lot Sizing Problem With Returns and Remanufacturing, *International Journal of Production Research*, 49 (9): 2519-2533.
- Schwarz, Christian (2010): *Effiziente Algorithmen zur optimalen Lösung von dynamischen Losgrößenproblemen mit integrierter Wiederaufarbeitung*, Logos Verlag: Berlin.
- Schwarz, Christian, Udo Buscher, and Steffen Rudert (2009): Dynamische Losgrößenplanung mit Wiederaufarbeitungs- und Entsorgungsoptionen, in: Stefan Voß, Julia Pahl, and Slavia Schwarze (eds.): *Logistik-Management: Systeme, Methoden, Integration*, Physica-Verlag: Berlin et al., 211-227.
- Teunter, Ruud H., Z. Pelin Bayindir, and Wilco van den Heuvel (2006): Dynamic Lot Sizing With Product Returns and Remanufacturing, *International Journal of Production Research*, 44 (20): 4377-4400.
- Van den Heuvel, Wilco (2004): On the Complexity of the Economic Lot-Sizing Problem With Remanufacturing Options, <http://people.few.eur.nl/wvandenheuvel/papers/ei200446.pdf> (Access date: 2011-09-12).
- Wagner, Harvey M. and Thomson M. Whitin (1958): Dynamic Version of the Economic Lot Size Model, *Management Science*, 5 (1): 8996.

Biography

Florian Sahling holds the position of an Assistant Professor for "Quantitative Supply Chain Management" at the Department of Production Management at the Leibniz Universität Hannover. He studied Mathematics and Management Science at the University of Hamburg and the University of Bath. He received his doctoral degree in 2009. His main area of research is production planning and scheduling.